CSE200 Lecture Notes
Simulating a TM with a Circuit

Lecture by Russell Impagliazzo
Notes by William Matthews

Lectures April 7, 2010 and April 12, 2010

1 Simulating a k-tape TM with a circuit

Given a k-tape TM M which runs in time at most T on inputs of size n, we wish to construct a circuit which simulates M on inputs of size n.

Let Σ and Q denote the tape alphabet and set of states of M respectively. We will begin by defining Boolean variables to encode the entire configuration of M at each time step. Each of these variables will correspond to either an input to the circuit, a constant, or the output of a gate.

- $w_{t,i,\ell,\sigma}$ for each $1 \leq t \leq T$, $1 \leq i \leq k$, $1 \leq \ell \leq T$, and $\sigma \in \Sigma$, where $w_{t,i,\ell,\sigma} = 1$ will correspond to having σ written on tape i in location ℓ at time step t.
- $h_{t,i,\ell}$ for each $1 \leq t \leq T$, $1 \leq i \leq k$, and $1 \leq \ell \leq T$, where $h_{t,i,\ell} = 1$ will correspond to having the tape head on the i^{th}tape at position ℓ at times step t.
- $s_{t,q}$ for $1 \leq t \leq T$ and $q \in Q$, where $s_{t,q} = 1$ corresponds to the TM being in state q at time step t.
- $z_{t,i,\sigma}$ for $1 \leq t \leq T$, $1 \leq i \leq k$, and $\sigma \in \Sigma$, where $z_{t,i,\sigma} = 1$ will correspond to having the symbol σ under the tape head on the i^{th}tape at time step t.
- $left_{t,i}, right_{t,i}$ for $1 \leq t \leq T$ and $1 \leq i \leq k$, where $left_{t,i} = 1$ and $right_{t,i} = 1$ correspond to the tape head moving left or right respectively when going from time step t to time step $t + 1$.

1.1 Time step 1

The variables $w_{1,1,\sigma,\ell}$, for $1 \leq \ell \leq n$, will be the input to the circuit, with $w_{1,1,\sigma,\ell} = 1$ when the ℓ^{th}symbol of the input is σ and 0 otherwise. For $i > 1$ or $\ell > n$, $w_{1,i,\sigma,\ell} = 1$ if an only if σ is the blank symbol. This corresponds to having the input string written on the first n cells of the first tape, and having blanks written everywhere else.

For all i, $h_{1,i,1} = 0$, and $h_{1,i,\ell} = 0$ for $\ell > 1$. This corresponds to having the each tape head on the first cell of its tape.

$s_{1,q_0} = 1$ and $s_{1,q} = 0$ for all $q \neq q_0$, where q_0 is the start state of M.

1.2 Subsequent time steps

For each time step t, tape i, and symbol σ, the tape head is over symbol σ if there is some ℓ such that the i^{th} head is in position ℓ and σ is written in position ℓ on the i^{th} tape.

$$z_{t,i,\sigma} = \bigvee_{\ell=1}^{T} (h_{t,i,\ell} \wedge w_{t,i,\ell,\sigma})$$

For each time step t and tape i, the movement of the tape head depends on the state q of the machine and the symbols $\sigma_1, \ldots, \sigma_k$ written under the tape heads on each tape.

$$left_{t,i} = \bigvee_{\substack{q,\sigma_1,\ldots\sigma_k \\ \text{such that } M \text{ moves head } i \text{ left} \\ \text{in state } q \text{ on symbols } \sigma_1,\ldots\sigma_k}} (s_{t,q} \wedge z_{t,1,\sigma_1} \wedge z_{t,2,\sigma_2} \wedge \cdots \wedge z_{t,k,\sigma_k})$$

$$right_{t,i} = \bigvee_{\substack{q,\sigma_1,\ldots\sigma_k \\ \text{such that } M \text{ moves head } i \text{ right} \\ \text{in state } q \text{ on symbols } \sigma_1,\ldots\sigma_k}} (s_{t,q} \wedge z_{t,1,\sigma_1} \wedge z_{t,2,\sigma_2} \wedge \cdots \wedge z_{t,k,\sigma_k})$$

For each time step t, tape i, location ℓ, and symbol σ, a σ is written on position ℓ of the i^{th} tape in two cases: The first case is if the tape head is not in location ℓ at time step $t-1$ and a σ is written there. The second case is if the tape head is in location ℓ at time step $t-1$ and the state of the machine q and symbols $\sigma_1, \ldots, \sigma_k$ under the k tape heads would cause a σ to be written.

$$\begin{aligned} w_{t,i,\ell,\sigma} &= (\neg h_{t-1,i,\ell} \wedge w_{t-1,i,\ell,\sigma}) \\ &\vee \left(h_{t-1,i,\ell} \wedge \bigvee_{\substack{q,\sigma_1,\ldots\sigma_k \\ \text{such that } M \text{ writes } \sigma \text{ on the } i^{\text{th}}\text{tape} \\ \text{in state } q \text{ on symbols } \sigma_1,\ldots\sigma_k}} (s_{t-1,q} \wedge z_{t-1,1,\sigma_1} \wedge z_{t-1,2,\sigma_2} \wedge \cdots \wedge z_{t-1,k,\sigma_k}) \right) \end{aligned}$$

For each time step t, tape i, and location ℓ, the tape head is in location ℓ on the i^{th} tape if it was in location ℓ at time $t-1$ and didn't move, or if it was in location $\ell+1$ and moved left, or if it was in location $\ell-1$ and moved right.

$$\begin{aligned} h_{t,i,\ell} &= (\neg left_{t-1,i} \wedge \neg right_{t-1,i} \wedge h_{t-1,i,\ell}) \\ &\vee (left_{t-1,i} \wedge h_{t-1,i,\ell+1}) \\ &\vee (right_{t-1,i} \wedge h_{t-1,i,\ell-1}) \end{aligned}$$

For each time step t and state q, whether the state of the TM is q depends only on the state q' and symbols $\sigma_1, \ldots, \sigma_k$ under the k tape heads at time $t-1$.

$$s_{t,q} = \bigvee_{\substack{q',\sigma_1,\ldots\sigma_k \\ \text{such that } M \text{ goes to state } q \\ \text{from state } q' \text{ on symbols } \sigma_1,\ldots\sigma_k}} (s_{t-1,q'} \wedge z_{t-1,1,\sigma_1} \wedge z_{t-1,2,\sigma_2} \wedge \cdots \wedge z_{t-1,k,\sigma_k})$$

The output of the circuit is 1 if the TM is in an accepting state at time T.

$$output = \bigvee_{\text{accepting } q} s_{T,q}$$

Starting with time step 1, we can convert these variables into a circuit: Each of the variables at time step 1 is either an input to the circuit, or a constant. For subsequent time steps, we have given formulas for the value of each variable in terms of previous variables. Thus, by constructing the circuit looking at increasing t, we can add gates to the circuit correspond to each variable. Finally, the output of the circuit is defined in terms of variables at time T.

1.3 Size of the circuit

Finally, in order to argue that circuits are a reasonable model of computation, we will show that the size of the circuit, is polynomial in the running time of the underlying TM M. To do so, we will look at the number of gates contributed by each type of variable multiplied by the number of variables of each type. Since k, $|Q|$, and $|\Sigma|$ are constants, we will hide their contributions in the big-O notation and focus on T. There are $O(T^2)$ h and w variables and each contributes $O(1)$ gates to the circuit. There are $O(T)$ z variables, and each contributes $O(T)$ gates, since there is an "or" over all T locations to compute each z variable. There are $O(T)$ s, $left$, and $right$ variables, and each contributes $O(1)$ gates (recall that k, $|Q|$, and $|\Sigma|$ are constant independent of T.) Adding all of these up (with a slight abuse of notation), we get $O(T^2) \times O(1) + O(T) \times O(T) + O(T) = O(T^2)$.

1.4 Uniform vs. non-uniform circuit families

Since a given circuit has a fixed number of inputs, to decide a language we need a family of circuits – one for each input size n. In general each circuit in such a family could be very different. A language L is in $P/poly$ if there exists a family of circuits that compute L and a constant k such that the size of the circuit for inputs of size n is at most $O(n^k)$. Note that $P \neq P/poly$ since every unary language (including undecidable ones) is in $P/poly$.

A family of circuits is uniform if there exists a TM M which on input 1^n will output the circuit in the family for inputs of size n. Furthermore, if the TM M runs in polynomial time, then the family of circuits is said to be P-uniform.

Given these definitions and the above conversion from TMs to circuit families it follows that every recursive/decidable language has a uniform family of circuits. In addition every language in P has a P-uniform family of circuits.